07-03-00

A

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of: Thomas A. Tetzlaff
Title: DIGITAL CIRCUIT SIMULATION
Attorney Docket No.: 884.279US1

PATENT APPLICATION TRANSMITTAL

BOX PATENT APPLICATION
Commissioner for Patents
Washington, D.C. 20231

We are transmitting herewith the following attached items and information (as indicated with an "X"):

X Return postcard.
X Utility Patent Application under 37 CFR § 1.53(b) comprising:
  X Specification (18 pgs, including claims numbered 1 through 25 and a 1 page Abstract).
  X Formal Drawing(s) (7 sheets).
  X Signed Combined Declaration and Power of Attorney (3 pgs).
  X Check in the amount of $936.00 to pay the filing fee.
X Assignment of the invention to Intel Corporation (2 pgs) and Recordation Form Cover Sheet.
X Check in the amount of $40.00 to pay the Assignment recording fee.
X Information Disclosure Statement (1 pgs), Form 1449 (1 pgs), and copies of cited references (3).

The filing fee has been calculated below as follows:

	No. Filed	No. Extra	Rate	Fee
TOTAL CLAIMS	25 - 20 =	5	x 18 =	$90.00
INDEPENDENT CLAIMS	5 - 3 =	2	x 78 =	$156.00
[ ] MULTIPLE DEPENDENT CLAIMS PRESENTED				$0.00
BASIC FEE				$690.00
TOTAL				$936.00

Please charge any additional required fees or credit overpayment to Deposit Account No. 19-0743.

SCHWEGMAN, LUNDBERG, WOESSNER & KLUTH, P.A.
P.O. Box 2938, Minneapolis, MN 55402 (612-373-6900)

By: Dana B. LeMoine
Atty: Dana B. LeMoine
Reg. No. 40,062

Customer Number 21186

"Express Mail" mailing label number: EL600376699US
Date of Deposit: June 30, 2000
I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner for Patents, Box Patent Application, Washington, D.C. 20231.

By: Shawn L. Hise
Signature:

DIGITAL CIRCUIT SIMULATION

INVENTOR

Thomas A. Tetzlaff

Schwegman, Lundberg, Woessner & Kluth, P.A.
1600 TCF Tower
121 South Eighth Street
Minneapolis, MN 55402
ATTORNEY DOCKET SLWK 884.279US1
Client Ref. No. P8537

DIGITAL CIRCUIT SIMULATION

Field

The present invention relates generally to the simulation of digital circuits, and more specifically to the switch-level simulation of digital circuits.

Background of the Invention

Switch-level simulators typically represent node values in a digital circuit as zero, one, and "unknown," and nodes are evaluated on a cycle-by-cycle basis. During each cycle, values present on input nodes to a block are used to evaluate output nodes of the block. For example, when simulating an inverter with a logical one on an input node, a switch-level simulator will evaluate the output node to a logical zero on the next cycle.

When a value of "unknown" is present on an input node, switch-level simulators typically evaluate the output node to a value of "unknown." This can be useful to alert a simulation user that a node has not been properly initialized, or that there is possibly a design flaw in the digital circuit being simulated.

Some digital circuits, during actual operation, will always evaluate an output node to the same value when a value of "unknown" is present on an input node. For example, when power is applied to a circuit, all node values are initially unknown, but many circuits always initialize nodes to the same value. These circuits, while having predictable behavior during actual operation, tend to propagate "unknown" node values during initialization of switch-level simulations.

One known method for initializing nodes in a simulation includes forcing node values to a known value. Such an approach is discussed in: A.J. van Genderen, "Network Initialization in a Switch-Level Simulator," Proc. IEEE European Design and Test Conference, p. 596, 1995. Another known method includes structural decomposition of sequential circuits. Such an approach is discussed in: Jalal A. Wehbeh and Daniel G. Saab, "On the Initialization of Sequential Circuits," Proc. IEEE International Test Conference, pp. 233-239, 1994. These and other prior art

initialization approaches typically remove unknown values from nodes at the beginning of simulation.

In some circuits, unknown values can occur not just at initialization, but also during simulation. Circuits with asynchronous loops that experience a node value of "unknown" during switch-level simulation typically propagate the unknown value throughout the asynchronous loop. Regardless of whether the circuit always recovers from the unknown state to the same known state during actual operation, the switch-level simulation typically tends to continue to propagate the unknown value throughout the asynchronous loop. As a result, the switch-level simulation may not accurately reflect the actual operation of the circuit being simulated.

For the reasons stated above, and for other reasons stated below which will be appreciated by those skilled in the art upon reading and understanding the present specification, there is a need in the art for an alternate method and apparatus to simulate circuits that can have unknown node values.

Brief Description of the Drawings

Figure 1 shows a digital circuit;

Figures 2A and 2B show a simulation model of an inverter and a function table therefor;

Figures 3A and 3B show diagrams of simulation models;

Figure 4 shows a table of simulation node states in a switch-level simulator;

Figures 5A-5C show an inverter function in a switch-level simulator;

Figure 6 shows a flowchart of a method of performing a simulation;

Figure 7 shows a diagram of a processing system; and

Figure 8 shows a memory that includes a data structure.

Description of Embodiments

In the following detailed description of the embodiments, reference is made to the accompanying drawings that show, by way of illustration, specific embodiments in which the invention may be practiced. In the drawings, like

numerals describe substantially similar components throughout the several views. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention. Other embodiments may be utilized and structural, logical, and electrical changes may be made without departing from the scope of the present invention. Moreover, it is to be understood that the various embodiments of the invention, although different, are not necessarily mutually exclusive. For example, a particular feature, structure, or characteristic described in one embodiment may be included within other embodiments. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims, along with the full scope of equivalents to which such claims are entitled.

The method and apparatus of the present invention provide a mechanism for simulating digital circuits in a switch-level simulator. In some embodiments, a switch-level simulator has three valid values for each node within a network of nodes: logical zero, logical one, and "unknown." For the purposes of this description, the "unknown" value is represented by the letter "X." When an X appears on an input node, it is propagated to an output node. Self-resetting circuits include an asynchronous loop that evaluates to a known value in the actual circuit, but that does not necessarily evaluate to a known value in switch-level simulators. The method and apparatus of the present invention provide an attribute to be included on a simulation model. The attribute is interpreted by the simulator, and the simulator conditionally treats an X on an input node as a known value. When the attribute is used on strategically placed models within a simulation of a self-resetting circuit, the correct simulation behavior is achieved.

Figure 1 shows a digital circuit. Digital circuit 100 includes P-channel transistors 102 and 104, N-channel transistor 106, and inverters 112 and 116. Digital circuit 100 is a generalized logic circuit, presented in simple form to aid in the discussion of the present invention. For example, circuit 100 is shown including a single N-channel transistor beneath the parallel P-channel transistors. With this single N-channel transistor, circuit 100 performs the function of an inverter. When

the voltage on logic input node 108 is high, the voltage present on logic output 110 is low. Likewise, when the voltage on logic input node 108 is low, the voltage present on logic output node 110 is high.

In practice, N-channel transistor 106 can be replaced with multiple transistors in series, in parallel, or in any combination thereof. For example, when two N-channel transistors are in series, a NAND gate is formed. When both input voltages are high, the output voltage will go low. Also for example, when two N-channel transistors are in parallel, a NOR gate is implemented. In the NOR gate example, the output presents a low voltage when either input voltage is high.

The N-channel transistors are collectively termed the "N-stack." The N-stack implements a logic function that, when satisfied, presents a low voltage on logic output node 110. In operation, the voltage on logic output node 110 of digital circuit 100 will go low when the N-stack provides a path to ground. P-channel transistors 102 and 104 are smaller in size than transistors in the N-stack, and so can be overdriven by the N-stack.

When logic output node 110 above the N-stack is pulled down to a low voltage by the N-stack, and then the path through the N-stack goes to a high impedance, the combination of P-channel transistors 102 and 104 and inverters 112 and 114 automatically cause the voltage on logic output node 110 to reset to a high voltage. Inverter 112 drives P-channel transistor 104 off, and inverter 116 drives P-channel transistor on, which pulls up logic output node 110 to a high voltage. After logic output node 110 transitions to a high voltage, inverter 112 drives P-channel transistor 114 on, and the circuit reaches steady state with logic output node 110 at a high voltage. The "self-resetting" action just described gives rise to the name "self-resetting circuit." The path through the inverter chain to P-channel transistor 102 and back to logic output 110 is termed the "self-reset path." After output node 110 is reset to to a high voltage, inverter 112 and P-channel transistor 104 act to maintain or "keep" the output high until some later time when the N-stack inputs cause node 110 to go low. This is termed the "keeper path."

Digital circuit 100 can be simulated in a switch-level simulator. Conventional switch-level simulators typically represent circuit components as switches that are open or closed, and as logic gates that drive logical values of zero, one, and X onto circuit nodes. A switch-level simulator can generally accurately simulate the self-resetting behavior described above with respect to the N-stack pulling the output node low and the self-resetting path pulling the output back high. Correctly evaluating the value of logic output node 110 when an X is in present in the circuit, however, can present a challenge. For example, when the simulator initializes the circuit (which is a simulation of the power-on sequence), the simulator attempts to evaluate logic states of the various nodes in the circuit. Logic output node 110 initially has a value of X. As previously described, the real circuit will evaluate the output to a high voltage because, even with an unknown value on the output, either the self-reset path or the keeper path is eventually active. Conventional switch-level simulators, however, do not model the phenomena that cause the real circuit to reach steady state. In this case, with X values on both of the P-channel transistor inputs, the simulator allows for the possibility that both transistors might be off, while in reality one or the other must eventually be on. The simulator has trouble initializing the self-resetting circuit in part because, an asynchronous loop is formed from logic output node 110, through the inverter chain that includes inverters 116 and 112, and through P-channel transistors 102 and 104.

Whenever logic output node 110 is driven to a low voltage, each node in the self-reset path, both in the real circuit and in a simulation, evaluate to a known state. When during the simulation, an input node to the N-stack presents an X, conventional switch-level simulators will propagate the X to logic output node 110. After the X on the N-stack input is driven with either a logical one or zero, the switch-level simulator will continue to propagate an X onto logic output node 110 because the asynchronous self-reset path will continue to evaluate to unknown values.

Attribute 114 on inverter 112 is an attribute on a simulation model that can be interpreted by a switch-level simulator. When the simulator detects an X on the

input node to inverter 112, the simulator checks the state of attribute 114. If the attribute is set, then the X on the inverter input is treated as a logical zero rather than an unknown state. Any simulation model within circuit 100 can have an attribute such as attribute 114. By placing attribute 114 on inverter 112, the behavior of the asynchronous self-reset loop can be modeled more accurately, in part because X's can be propagated to the logic output node 110 by the N-stack, but will not be unnecessarily held there by the self-reset circuit. As soon as the N-stack evaluates to a known value, logic output node 110 also evaluates to a known value. In other words, when the real circuit behavior will always reset logic output node 110 to a high voltage, the switch level simulator will always evaluate logic output node 110 to a logical one.

Attribute 114 allows the switch level simulator to initialize the circuit correctly, as well as more accurately model the behavior of the circuit when an X is present in the circuit during an on-going simulation. The inverter chain that includes inverter 112 and 116 can include additional inputs for purposes such as clock gating, and clock stopping. The placement of attribute 114 on inverter 112 allows these additional inputs to continue to function correctly.

Figure 2A shows a diagram of a simulation model of an inverter. Inverter model 202 has attribute 206 associated therewith. Input node 204 is an input to inverter model 202, which produces a value on output node 208. When a logical zero appears on input node 204, a simulator will evaluate output node 208 to a logical one. Likewise, when a logical one appears on input node 204, the simulator will evaluate output node 208 to a logical zero. When an X appears on input node 204, the switch-level simulator will propagate the X to output node 208. Attribute 206 can be utilized to alter the behavior of the switch-level simulator such that the X on input node 204 is not necessarily propagated to output node 208. In some embodiments, attribute 206 is a Boolean variable capable of having one of two states. In other embodiments, attribute 206 is a multi-valued variable.

Figure 2B shows a function table for the simulation model shown in Figure 2A. Entries 252 and 254 show the normal operation of an inverter. For example,

entry 252 shows that the output can be evaluated to a logical one when the input is at a logical zero. Entry 254 shows that the output can be evaluated to a logical zero when the input is at a logical one. Entries 256 and 258 show the operation of inverter model 202 (Figure 2A) when an X is present on input node 204.

When attribute 206 is clear, as shown in entry 256, the switch-level simulator propagates an X from input node 204 to output node 208. In contrast, when attribute 206 is set, the switch-level simulator treats an X on input node 204 as a logical zero, and evaluates output node 208 to a logical one. In the embodiment shown in Figure 2B, attribute 206 is a Boolean variable capable of being either clear or set. When set, an unknown value on an input node is treated as a logical zero. In other embodiments, an unknown value on an input node can be treated as a logical one. In still other embodiments, where attribute 206 is a multi-valued variable, an unknown value on an input node can be treated as either a logical zero or a logical one. Attribute 206 is shown as an attribute of inverter model 202.

Figure 3A shows a diagram of a simulation model of a logic function. Simulation model 300 includes logic function 312 having input nodes 302 and 306, and output node 310. Logic function 312 can be any logic function capable of being simulated using a switch-level simulator. Examples include AND gates, OR gates, NAND gates, NOR gates, and the like. The switch-level simulator evaluates a value on output node 310 as a function of the values on input nodes 302 and 306. Input node 302 has attribute 304 associated therewith. Likewise, input node 306 has attribute 308 associated therewith. Each of attributes 304 and 308 affects how a switch-level simulator treats an unknown input value on nodes 302 and 306 respectively. For example, in some embodiments, when attribute 304 is set, a switch-level simulator will interpret an X on node 302 as a logical zero. In other embodiments, when attribute 304 is set, the switch-level simulator will interpret an X on node 302 as a logical one. Attribute 308 functions in the same way, and affects the switch-level simulator's behavior when an X is present on input node 306. In the embodiment of Figure 3A, attributes 304 and 308 are attributes of input nodes, rather than attributes of logic function 312. This is in contrast to the embodiment shown in

Figure 2A, where attribute 206 is an attribute of inverter model 202, rather than an attribute of input node 204. It can be seen, therefore, that the method and apparatus of the present invention contemplate attributes associated with different types of entities in a simulation.

Figure 3B shows a diagram of a simulation model of a transistor. Simulation model 350 is a model of a field effect transistor (FET). As used in a switch-level simulator, simulation model 350 appears as a switch. When the switch is closed, drain node 354 is coupled to source node 356, and when the switch is open, drain node 354 is not coupled to source node 356. The switch is closed when gate node 352 has a logical one imposed thereon, and is open when gate node 352 has a logical zero imposed thereon. Attribute 358 is associated with model 350. Like attributes discussed with reference to previous figures, attribute 358 can affect how a switch-level simulator interprets a value of X on a node. For example, if gate node 352 has a logical one imposed thereon, and attribute 358 is set, and drain node 354 has an X, the switch-level simulator can interpret the X as either a logical zero or logical one, and can propagate that value to source node 356. In the embodiment shown in Figure 3B, attribute 358 is associated with the body of simulation model 350. In other embodiments, attributes are associated with each node, such as drain node 354 and source node 356.

Figure 4 shows a table of simulation node states in a switch-level simulator. As shown in table 400, a switch-level simulator can represent node values of logical zero, logical one, and X, as a two bit word. The two bit word shown in table 400 includes two bits labeled HIGH_BIT and LOW_BIT. A logical zero is represented as a "01" pattern in the two bit word, a logical one is represented as a "10" pattern in the two bit word, and an X is represented as a "11" pattern. Table 400 shows one possible encoding for a logical zero, logical one, and X using a two bit word. Other encodings can be used without departing from the scope of the present invention. For example, an X can be represented as a "00" pattern and the patterns used for logical one and logical zero could be interchanged without departing from the scope of the present invention.

Figures 5A-5C show an inverter function in a switch-level simulator. Figure 5A shows an input node value 502, output node value 506, and the operational mechanism 504 of an inverter simulation model. Input node value 502 has a HIGH_BIT value of one, and a LOW_BIT value of zero. As shown in table 400 (Figure 4), this corresponds to a logical one on the input node to the inverter. Mechanism 504, which models an inverter, when evaluating output node value 506 from input node value 502, swaps the values of the HIGH_BIT and LOW_BIT. In some embodiments, the simulator implements a "MOV" operation that moves the HIGH_BIT to the LOW_BIT and vice versa. In the example of Figure 5A, this creates a HIGH_BIT value of zero, and a LOW_BIT value of one on output node value 506. This corresponds to an output value of logical zero, which is the expected result from an inverter having an input value of logical one.

Figure 5B shows the operation of the same inverter model having an X on the input node. Input node value 522 has a two bit word value of "11," which corresponds to an X. When inverter model mechanism 504 swaps the HIGH_BIT and the LOW_BIT, the X of input node value 522 is propagated to output node value 526. Again, this is the expected result in a switch-level simulator evaluating an output node value on an inverter having an unknown value on an input node.

Figure 5C shows the action of an inverter model having an X on an input node, and having an attribute set. Input node value 532 is an X. This corresponds to an inverter having a value of X imposed on an input node. When the attribute is not set, the action is the same as that shown in Figure 5B. That is to say, the X on the input node is propagated to the output node. When the attribute is set, however, the action of Figure 5C takes place. The switch-level simulator treats an X on the input node of the inverter model as a logical zero, to produce a logical one on the output. In some embodiments, this is accomplished by inverting one bit of the two bit word representing input node value 532. Action 534 implements a "NOT" function to invert the HIGH_BIT of the two bit word to create a temporary logical zero, shown at temporary node value 536. The normal operation of the inverter model is then

carried out having a logical zero as a temporary input value, and a logical one is produced on the output node as output node value 540.

Figures 5A-5C describe an example embodiment for treating an X on an input node of an inverter as a value other than X. The inverter mechanism is accomplished using the two bit word node representation by swapping bits in the word using a MOV operation. The method and apparatus of the present invention also apply to other types of circuits, such as more complex logic gates. More complex logic gates can utilize AND, OR, and other operations in addition to the MOV operations of the inverter model while still practicing the present invention. Regardless of the mechanism of the simulation model, the method and apparatus of the present invention can aid in the resolution of node values in the presence of unknown node values.

Figure 6 shows a method of evaluating the value of a node in a simulation. Method 600 begins in decision block 610 where a determination is made whether an input node has a value of X. If not, method 600 continues with block 660, where the value on the output node is evaluated as a function of the inputs. For example, if the node being evaluated is the output node of an inverter, and the input node has a value of one, the output node will be assigned a value of zero. If, in decision block 610, it is determined that an input node has a value of X, method 600 continues in block 620.

In block 620, an attribute is accessed by the simulator. The attribute of block 620 can be an attribute on the input node having an X, or can be an attribute associated with the simulation model itself. The simulator determines whether the attribute is set in decision block 630. If the attribute is not set, method 600 continues with block 640, where the node is evaluated as if the X is present in the input node. This can include propagating the value of X to the node being evaluated, or can include evaluating the output node to a known value. For example if the gate is an inverter with an X on the input node, the X will be propagated to the output. If, on the other hand, the gate is a two input NAND gate with the other input node having a value of 0, the output node can be evaluated to a value of 1. If the attribute is set,

method 600 transitions from decision block 630 to block 650 where the value of X on the input node is treated as a logical zero. Using the logical zero as the value on input node, method 600 continues with block 660 where the output node is evaluated. In the embodiment described in method 600, an unknown value is treated as a logical zero when the attribute is set. In other embodiments, the unknown value can be treated as a logical one.

Figure 7 shows a processing system. Processing system 700 includes processor 720 and memory 730. In some embodiments, processor 720 represents a computer that performs a simulation of a circuit such as circuit 100 (Figure 1). In some embodiments, processor 720 is a processor capable of executing software embodiments of methods, such as that shown in Figure 6. Processing system 700 can be a personal computer (PC), mainframe, handheld device, portable computer, set-top box, or any other system that includes software.

Memory 730 represents an article that includes a machine readable medium. For example, memory 730 represents any one or more of the following: a hard disk, a floppy disk, random access memory (RAM), read only memory (ROM), flash memory, CDROM, or any other type of article that includes a medium capable of holding information readable by processor 720. Memory 730 can store instructions for performing the execution of the various method embodiments of the present invention. Memory 730 can also be a memory for holding a data structure such as the data structure embodiment shown in Figure 8.

Figure 8 shows a memory holding a data structure describing a simulation model. Memory 800 includes data structure 810, which in turn includes simulation model characteristics 820 and attribute 830. Memory 800 can be any type memory capable of holding data structure 810. Examples include memory within a computer and media types useful for distribution. For example, memory 800 can be a CDROM or other disk holding data structure 810 along with other data structures that form a library of simulation models.

Data structure 810 includes simulation model characteristics 820 and attribute 830. Simulation model characteristics 820 include information describing the

simulation model for use in simulation. For example, referring now back to Figure 3A, a data structure entry describing logic function 312 would be included in simulation model characteristics 820. If logic function 312 were an AND gate, the AND gate logical function would be described in simulation model characteristics 820. Attribute 830 corresponds to attributes 304 and 308. When the simulation model described by data structure 810 includes multiple input nodes, multiple attributes 830 can be included, one associated with each input node. Alternatively, attribute 830 can be a single attribute associated with the entire simulation model rather than a particular input.

The method and apparatus of the present invention provide a mechanism for evaluating node values in a switch-level simulation in the presence of unknown input node values. In a self-resetting circuit, where the real circuit behavior will always evaluate the output node to a logical one after a path to ground has been removed, switch-level simulation can be problematic. The method and apparatus of the present invention can faithfully simulate the correct behavior of self-resetting circuits.

It is to be understood that the above description is intended to be illustrative, and not restrictive. Many other embodiments will be apparent to those of skill in the art upon reading and understanding the above description. The scope of the invention should, therefore, be determined with reference to the appended claims, along with the full scope of equivalents to which such claims are entitled.

What is claimed is:

1. A computer-implemented method of initializing a simulation comprising:

accessing an attribute on a simulation model; and

responsive to the attribute, conditionally treating an unknown state of an input node on the simulation model as a known state.

2. The computer-implemented method of claim 1 wherein conditionally treating an unknown state comprises:

when the attribute is set, evaluating an output node of the simulation model using the known state for the input node rather than the unknown state.

3. The computer-implemented method of claim 2 wherein evaluating an output node comprises detecting an X on the input node, and evaluating the output node as if a 0 was on the input node.

4. The computer-implemented method of claim 3 wherein the simulation model is a model of an inverter, and evaluating an output node comprises detecting an X on the input node, and evaluating the output node to a 1.

5. The computer-implemented method of claim 2 wherein evaluating an output node comprises detecting an X on the input node, and evaluating the output node as if a 1 was on the input node.

6. The computer-implemented method of claim 5 wherein the simulation model is a switch-level model of a transistor, and evaluating an output node comprises detecting an X on a gate node of the switch-level model of the transistor, and logically closing the switch-level model of the transistor.

7. A computer-implemented method of evaluating an output node of a device model comprising:

determining whether an input node of the device model has an unknown value assigned thereto; and

responsive to an attribute on the device model, if the input node has an unknown value, conditionally evaluating the output node as if the input node had a known value.

8. The computer-implemented method of claim 7 wherein the attribute has at least two valid states comprising a first state signifying that the unkown value should be propagated to the output node, and a second state signifying that the unknown state should be treated as a known state, and wherein conditionally evaluating comprises:

when the attribute is in the first state and the input node has an unknown value, propagating the unknown value to the output node.

9. The computer-implemented method of claim 8 further comprising:

when the attribute is in the second state and the input node has an unknown value, evaluating the output node as if the input node had a known value thereon.

10. The computer-implemented method of claim 7 wherein the method is performed within a three value simulator that represents node values by a two bit word, and wherein evaluating the output node comprises:

inverting one bit of the two bit word representing the unknown value on the input node to create a temporary two bit word; and

evaluating the output node using the temporary two bit word as the input node value.

11. A computer-implemented method of simulating a self-resetting circuit, the method comprising:

detecting the presence of an X value on a node within the self-resetting circuit;

accessing an attribute on a device model that represents a device within the self-resetting circuit, wherein the node having the X value is an input node of the device; and

when the attribute is set, simulating the self-resetting circuit as if the node had a non-X value.

12. The computer-implemented method of claim 11 wherein simulating the self-resetting circuit as if the node had a non-X value comprises simulating the self-resetting circuit as if the node had a value of zero.

13. The computer-implemented method of claim 12 wherein the device comprises an inverter, and simulating comprises evaluating an output node of the inverter to a value of one.

14. The computer-implemented method of claim 11 wherein the X value is represented by a two bit word, and simulating the self-resetting circuit as if the node had a non-X value comprises inverting at least one bit of the two bit word.

15. The computer-implemented method of claim 14 wherein the device model represents a first inverter in an inverter chain, and simulating the self-resetting circuit further comprises exchanging the values of the bits in the two bit word to create a two bit word that represents the value of an output node of the device model.

16. An article having a computer readable medium, the computer readable medium having instructions stored thereon for performing a method of initializing a device model in a simulation, the method comprising:

accessing an attribute of the device model to ascertain a state of the attribute; and

responsive to the state of the attribute, conditionally treating an X on an input node of the device model as a value other than an X.

17. The article of claim 16 wherein the attribute is associated with the input node, the method further comprising:

accessing a second attribute of the device model, the second attribute being associated with a second input node; and

responsive to a state of the second attribute, conditionally treating an X on the second input node as a value other than an X.

18. The article of claim 16 wherein conditionally treating comprises:

conditionally treating an X on any input node of the device model as a value other than an X.

19. The article of claim 16 wherein the device model is a model of an inverter, the simulation is a switch-level simulation, and conditionally treating comprises:

when the attribute is set and an X is present on an input node to the model of the inverter, evaluating an output node of the model of the inverter to a 1.

20. The article of claim 16 wherein the device model is a model of an inverter, the simulation is a switch-level simulation, and conditionally treating comprises:

when the attribute is set and an X is present on an input node to the model of the inverter, evaluating an output node of the model of the inverter to a 0.

21. An article having a computer readable medium, the computer readable medium comprising a data structure describing a device model for use in a simulator, the data structure comprising an attribute to signify whether an X on an input node of the device model should be treated as a value other than X during a simulation.

22. The article of claim 21 wherein the attribute is associated with the device model such that the attribute is a single attribute configured to affect a simulator's behavior relative to all input nodes of the device model.

23. The article of claim 21 wherein the attribute is associated with the input node of the device model such that the attribute is configured to affect a simulator's behavior relative to only the input node to which the attribute is associated.

24. The article of claim 23 wherein the data structure further comprises a second attribute associated with a second input node of the device model, the second attribute being configured to signify whether an X on the second input node of the device model is to be treated a value other than X during simulation.

25. The article of claim 21 wherein the device model represents an inverter, and the device model is configured to be used in a switch-level simulator.

Abstract of the Disclosure

A self-resetting circuit is simulated in a switch-level simulator using simulation models that can conditionally treat an unknown value on an input node as a known value. An attribute is included with the simulation model. The attribute specifies to the simulator whether to treat an unknown value as a logical zero or a logical one. A single attribute can be associated with the simulation model, or one attribute can be associated with each input node. Self-resetting circuits can be simulated from an initial state that includes unknown states. The proper logical initialization behavior can be simulated while still allowing the self-resetting circuit to propagate unknown states during normal operation and simulation.

"Express Mail" mailing label number: EL600376699US
Date of Deposit: June 30, 2000
I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Assistant Commissioner for Patents, Washington, D.C. 20231
Printed Name: Shawn Hise
Signature:

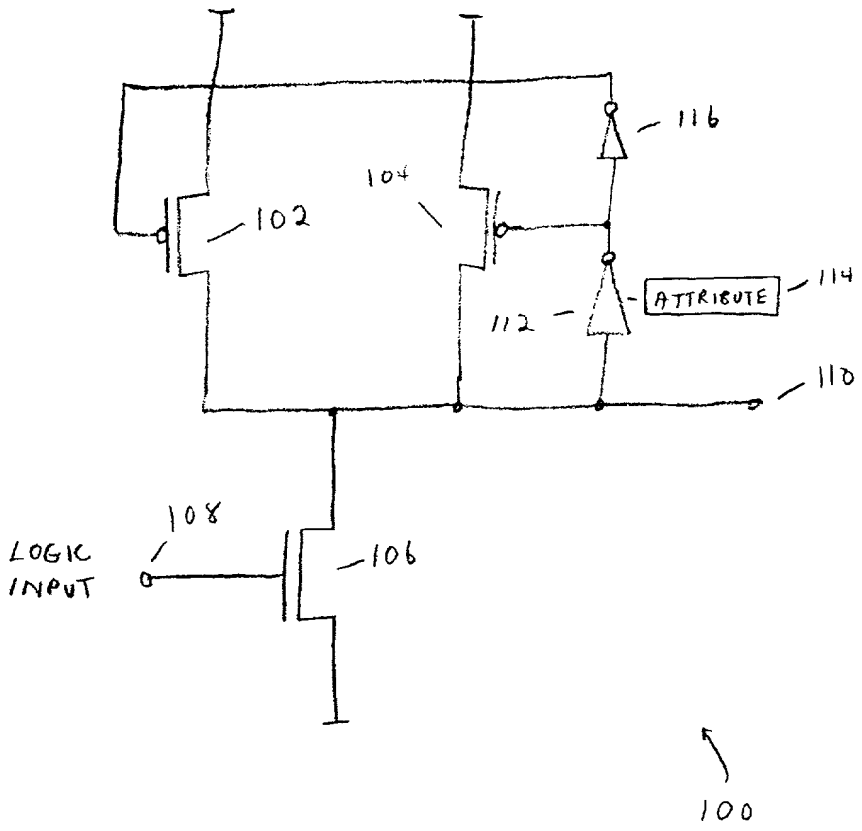
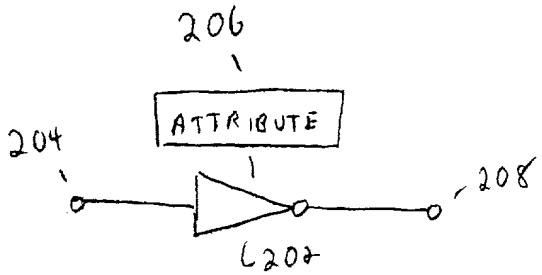


FIG. 1

884.279 vsl



200

FIG 2A

	INPUT	ATTRIBUTE	OUTPUT
252	0	—	1
254	1	—	0
256	X	CLEAR	X
258	X	SET	1

250

FIG. 2B

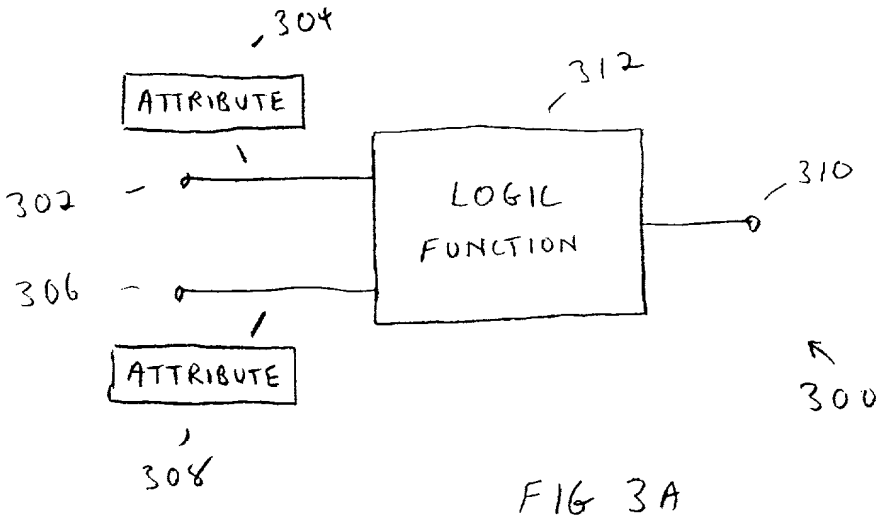

304
ATTRIBUTE
312
302
LOGIC
FUNCTION
310
306
ATTRIBUTE
308
300

FIG 3A

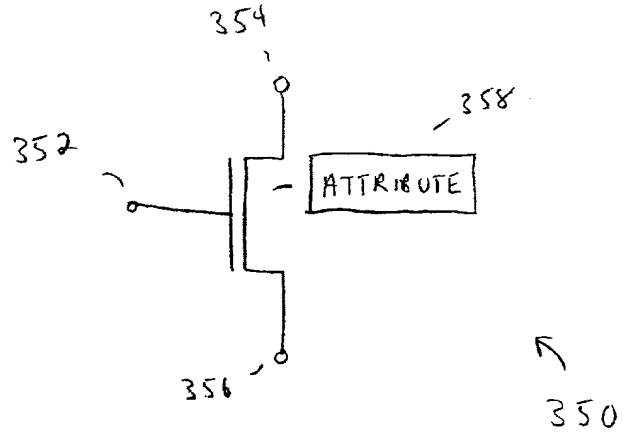

354
358
352
ATTRIBUTE
356
350

FIG 3B

	"0"	"1"	"X"
HIGH-BIT	0	1	1
LOW-BIT	1	0	1

← 400

FIG. 4

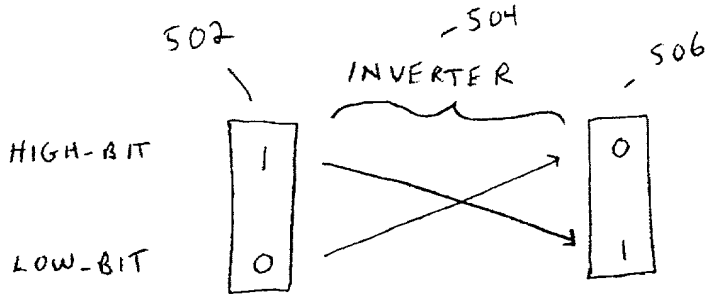


FIG 5A

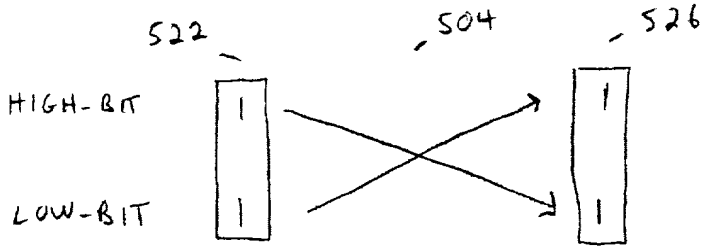


FIG 5B

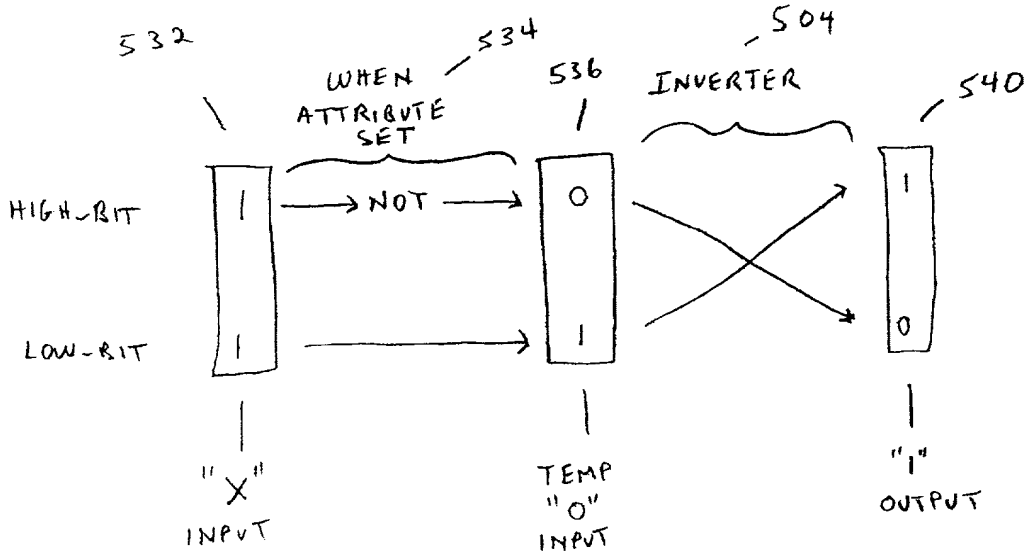


FIG 5C

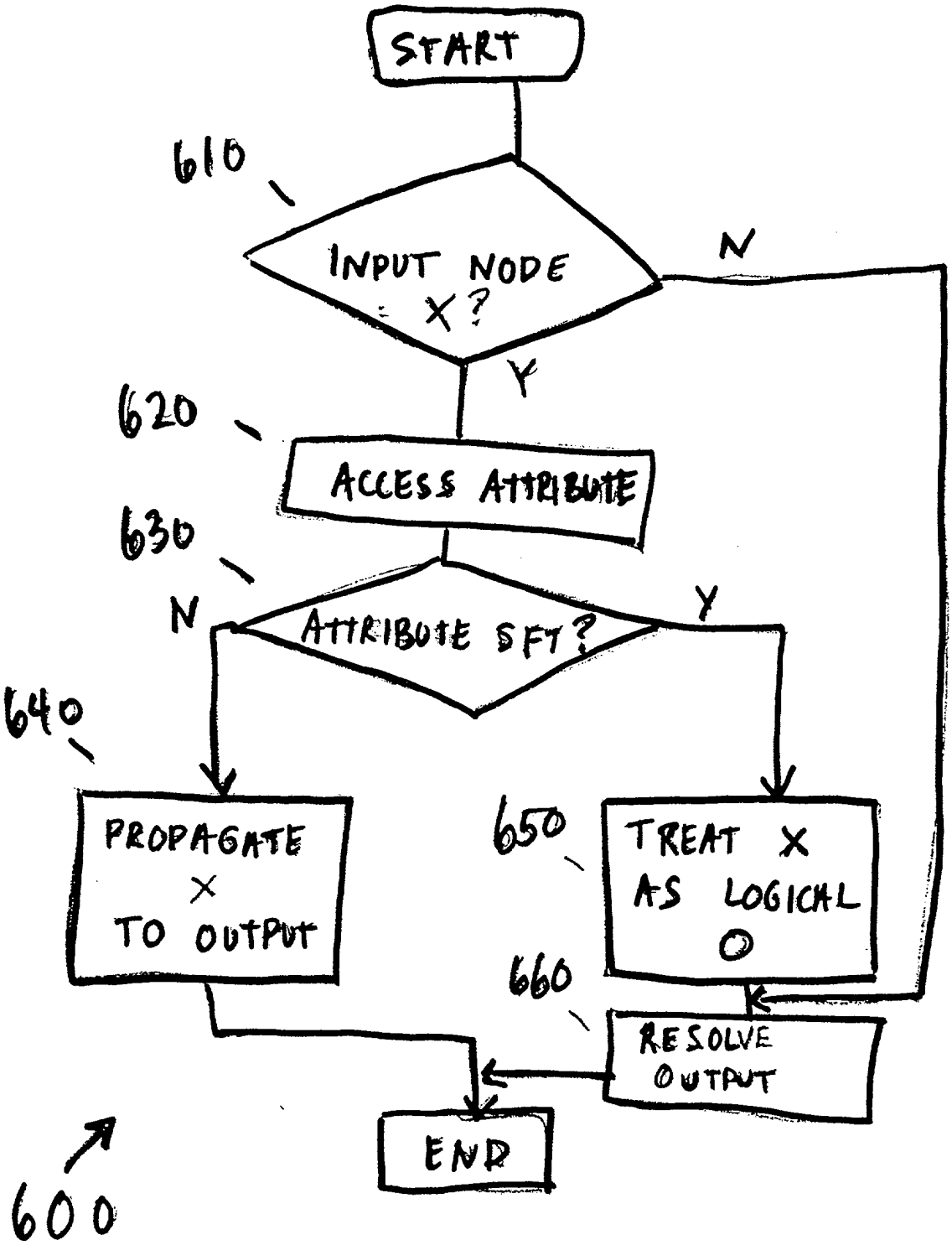

START
610
INPUT NODE = X?
N
Y
620
ACCESS ATTRIBUTE
630
ATTRIBUTE SET?
N
Y
640
PROPAGATE X TO OUTPUT
650
TREAT X AS LOGICAL 0
660
RESOLVE OUTPUT
END
600

FIG 6

FIG. 7

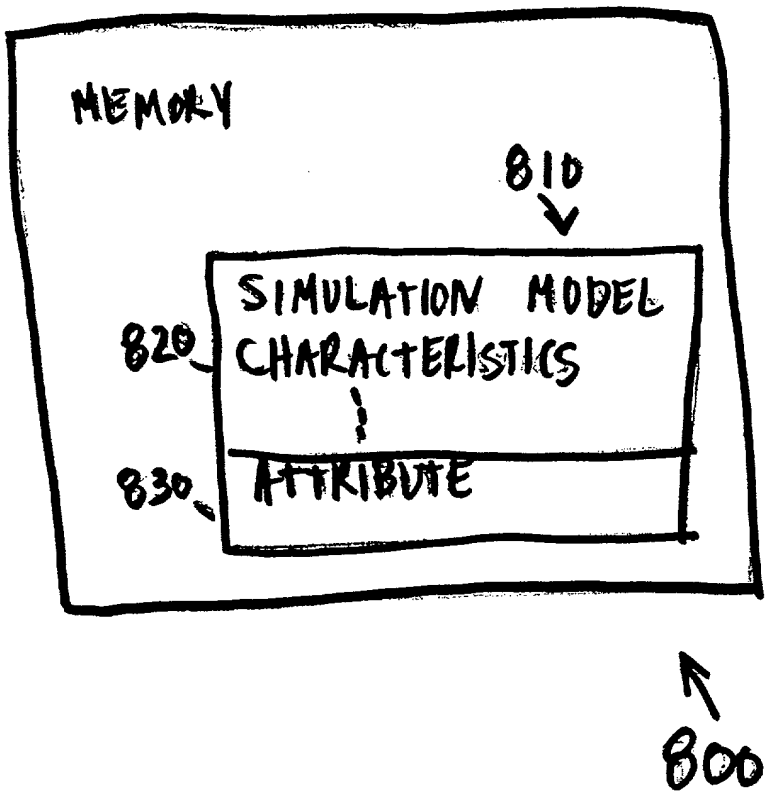

MEMORY
810
SIMULATION MODEL CHARACTERISTICS
820
ATTRIBUTE
830
800

FIG. 8

Attorney Docket No. 884.279US1

SCHWEGMAN ■ LUNDBERG ■ WOESSNER ■ KLUTH

United States Patent Application

COMBINED DECLARATION AND POWER OF ATTORNEY

As a below named inventor I hereby declare that: my residence, post office address and citizenship are as stated below next to my name; that

I verily believe I am the original, first and sole inventor of the subject matter which is claimed and for which a patent is sought on the invention entitled: **DIGITAL CIRCUIT SIMULATION**.

The specification of which is attached hereto.

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with 37 C.F.R. § 1.56 (attached hereto). I also acknowledge my duty to disclose all information known to be material to patentability which became available between a filing date of a prior application and the national or PCT international filing date in the event this is a Continuation-In-Part application in accordance with 37 C.F.R. §1.63(e).

I hereby claim foreign priority benefits under 35 U.S.C. §119(a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on the basis of which priority is claimed:

No such claim for priority is being made at this time.

I hereby claim the benefit under 35 U.S.C. § 119(e) of any United States provisional application(s) listed below:

No such claim for priority is being made at this time.

I hereby claim the benefit under 35 U.S.C. § 120 or 365(c) of any United States and PCT international application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT international application in the manner provided by the first paragraph of 35 U.S.C. § 112, I acknowledge the duty to disclose material information as defined in 37 C.F.R. § 1.56(a) which became available between the filing date of the prior application and the national or PCT international filing date of this application:

No such claim for priority is being made at this time.

Attorney Docket No.: 884.279US1
DIGITAL CIRCUIT SIMULATION
Filing Date: Even Date Herewith

I hereby appoint the following attorney(s) and/or patent agent(s) to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith:

Aldous, Alan K.	Reg. No. 31,905	Kalis, Jamal M.	Reg. No. 37,650	Park, Ellen	Reg. No. 34,055
Anglin, J. Michael	Reg. No. 24,916	Kalson, Seth Z.	Reg. No. 40,670	Parker, J. Kevin	Reg. No. 33,024
Bianchi, Timothy E.	Reg. No. 39,610	Kaplan, David J.	Reg. No. 41,105	Perdok, Monique M.	Reg. No. 42,989
Billion, Richard E.	Reg. No. 32,836	Kaufmann, John D.	Reg. No. 24,017	Prout, William F.	Reg. No. 33,995
Black, David W.	Reg. No. 42,331	Klima-Silberg, Catherine I.	Reg. No. 40,052	Reynolds, Thomas C.	Reg. No. 32,488
Brake, R. Edward	Reg. No. 37,784	Kluth, Daniel J.	Reg. No. 32,146	Schumm, Sherry W.	Reg. No. 39,422
Brennan, Leonicde M.	Reg. No. 35,832	Lacy, Rodney L.	Reg. No. 41,136	Schwegman, Micheal L.	Reg. No. 25,816
Brennan, Thomas F.	Reg. No. 35,075	Lam, Peter	Reg. No. 44,855	Seddon, Kenneth M.	Reg. No. 43,105
Brooks, Edward J., III	Reg. No. 40,925	Lemaire, Charles A.	Reg. No. 36,198	Seeley, Mark	Reg. No. 32,299
Burge, Ben	Reg. No. 42,372	LeMoine, Dana B.	Reg. No. 40,062	Skabrat, Steven P.	Reg. No. 36,279
Chu, Dinh C.P.	Reg. No. 41,676	Lundberg, Steven W.	Reg. No. 30,568	Skaist, Howard A.	Reg. No. 36,008
Clark, Barbara J.	Reg. No. 38,107	Mack, Lisa K.	Reg. No. 42,825	Smith, Michael G.	Reg. No. 45,368
Dahl, John M.	Reg. No. 44,639	Maeyaert, Paul L.	Reg. No. 40,076	Speier, Gary J.	Reg. No. 45,458
Drueger, Jeffrey S.	Reg. No. 41,000	Malki, Peter C.	Reg. No. 42,832	Steffey, Charles E.	Reg. No. 25,179
Drake, Eduardo E.	Reg. No. 40,594	Malen, Peter L.	Reg. No. 44,894	Su, Gene I.	Reg. No. 45,140
Embretson, Janet E.	Reg. No. 39,665	Mates, Robert E.	Reg. No. 35,271	Terry, Kathleen R.	Reg. No. 31,884
Faatz, Cynthia Thomas	Reg. No. 39,973	McCrackin, Ann M.	Reg. No. 42,858	Tong, Viet V.	Reg. No. 45,416
Fordenbacher, Paul J.	Reg. No. 42,546	Mirho, Charles A.	Reg. No. 41,199	Viksnins, Ann S.	Reg. No. 37,748
Forrest, Bradley A.	Reg. No. 30,837	Nama, Kash	Reg. No. 44,255	Wells, Calvin E.	Reg. No. 43,256
Gamon, Owen J.	Reg. No. 36,143	Nelson, Albin J.	Reg. No. 28,650	Werner, Raymond J.	Reg. No. 34,752
Greaves, John N.	Reg. No. 40,362	Nielsen, Walter W.	Reg. No. 25,539	Winkle, Robert G.	Reg. No. 37,474
Harris, Robert J.	Reg. No. 37,346	Novakoski, Leo V.	Reg. No. 37,198	Woessner, Warren D.	Reg. No. 30,440
Huebsch, Joseph C.	Reg. No. 42,673	Oh, Allen J.	Reg. No. 42,047	Young, Charles K.	Reg. No. 39,435
Jurkovich, Patti J.	Reg. No. 44,813	Padys, Danny J.	Reg. No. 35,635		

I hereby authorize them to act and rely on instructions from and communicate directly with the person/assignee/attorney/firm/organization/who/which first sends/sent this case to them and by whom/which I hereby declare that I have consented after full disclosure to be represented unless/until I instruct Schwegman, Lundberg, Woessner & Kluth, P.A. to the contrary.

Please direct all correspondence in this case to **Schwegman, Lundberg, Woessner & Kluth, P.A.** at the address indicated below:

P.O. Box 2938, Minneapolis, MN 55402
Telephone No. (612)373-6900

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of sole inventor: **Thomas A. Tetzlaff**
Citizenship: **United States of America**
Residence: **Portland, OR**
Post Office Address: 2566 NW Marshall Street
#B3
Portland, OR 97210

Signature: Thomas A. Tetzlaff Date: June 30, 2000
Thomas A. Tetzlaff

Full Name of inventor:
Citizenship:
Residence:
Post Office Address:

Signature: _____ Date: _____

§ 1.56 Duty to disclose information material to patentability.

(a) A patent by its very nature is affected with a public interest. The public interest is best served, and the most effective patent examination occurs when, at the time an application is being examined, the Office is aware of and evaluates the teachings of all information material to patentability. Each individual associated with the filing and prosecution of a patent application has a duty of candor and good faith in dealing with the Office, which includes a duty to disclose to the Office all information known to that individual to be material to patentability as defined in this section. The duty to disclose information exists with respect to each pending claim until the claim is canceled or withdrawn from consideration, or the application becomes abandoned. Information material to the patentability of a claim that is canceled or withdrawn from consideration need not be submitted if the information is not material to the patentability of any remaining under consideration in the application. There is no duty to submit information which is not material to the patentability of any existing claim. The duty to disclose all information known to be material to patentability is deemed to be satisfied if all information known to be material to patentability of any claim issued in a patent was cited by the Office or submitted to the Office in the manner prescribed by §§ 1.97(b)-(d) and 1.98. However, no patent will be granted on an application in connection with which fraud on the Office was practiced or attempted or the duty of disclosure was violated through bad faith or intentional misconduct. The Office encourages applicants to carefully examine:

(1) prior art cited in search reports of a foreign patent office in a counterpart application, and

(2) the closest information over which individuals associated with the filing or prosecution of a patent application believe any pending claim patentably defines, to make sure that any material information contained therein is disclosed to the Office.

(b) Under this section, information is material to patentability when it is not cumulative to information already of record or being made of record in the application, and

(1) It establishes, by itself or in combination with other information, a prima facie case of unpatentability of a claim; or

(2) It refutes, or is inconsistent with, a position the applicant takes in:

(i) Opposing an argument of unpatentability relied on by the Office, or

(ii) Asserting an argument of patentability.

A prima facie case of unpatentability is established when the information compels a conclusion that a claim is unpatentable under the preponderance of evidence, burden-of-proof standard, giving each term in the claim its broadest reasonable construction consistent with the specification, and before any consideration is given to evidence which may be submitted in an attempt to establish a contrary conclusion of patentability.

(c) Individuals associated with the filing or prosecution of a patent application within the meaning of this section are:

(1) Each inventor named in the application:

(2) Each attorney or agent who prepares or prosecutes the application; and

(3) Every other person who is substantively involved in the preparation or prosecution of the application and who is associated with the inventor, with the assignee or with anyone to whom there is an obligation to assign the application.

(d) Individuals other than the attorney, agent or inventor may comply with this section by disclosing information to the attorney, agent, or inventor.